Appl. No. : 09/523,877
Filed : March 13, 2000

IN THE CLAIMS

Please cancel Claims 23, 35, and 41 without prejudice, and amend Claims 1, 2, 14, 17, 20, 37, 38, 39, 40, 42, and 43 as follows:

1. (Currently amended) A method of controlling the execution of instructions within a pipelined processor, comprising:

providing an instruction set comprising a plurality of instruction words, each of said instruction words comprising a plurality of data bits, at least one of said words comprising a jump instruction having at least one user-configurable mode and at least one user-definable mode associated therewith, said user-configurable and user-definable modes each being specified by the same ones of said plurality of data bits, said at least one user-definable mode not being predetermined in terms of function;

assigning one of a plurality of values to said ones of said data bits of said at least one jump instruction; and

controlling the execution of at least one subsequent instruction within said pipeline based on said one assigned value of said ones of data bits when said at least one jump instruction is decoded.

2. (Currently amended) The method of Claim 1, wherein the act of assigning comprises:

identifying ~~a plurality~~ said ones of data bits within said at least one jump instruction; and

assigning one of two discrete values to each of said ones of data bits, the combination of said two discrete values representing at least three jump delay slot modes within said processor.

3. (Previously presented) The method of Claim 2, wherein the act of controlling the execution based on said discrete values comprises selecting at least one mode from the group comprising:

(i) executing said at least one subsequent instruction under all circumstances;

(ii) executing said at least one subsequent instruction only if a jump occurs; and

(iii) stalling the pipeline or inserting a bubble into the pipeline if a jump occurs.

4. (Previously presented) The method of Claim 3, wherein said at least one jump instruction comprises a conditional branch instruction.

5. (Previously presented) The method of Claim 1, wherein the act of controlling the execution based on said one assigned value comprises:

(i) executing said at least one subsequent instruction under all circumstances;

(ii) executing said at least one subsequent instruction only if a jump occurs; and

(iii) stalling the pipeline or inserting a bubble into the pipeline if a jump occurs.

6.-13. (Cancelled)

14. (Currently amended) A digital processor comprising:

a processor core having a multistage instruction pipeline, said core being adapted to decode and execute an instruction set comprising a plurality of instruction words;

a data interface between said processor core and an information storage device; and

an instruction set comprising a plurality of instruction words, at least one of said instruction words being a user-configurable jump instruction containing data defining a plurality of jump delay slot modes and at least one user-defined mode, said jump delay slot modes and at least one user-defined mode each being specified by the same portions of said data, said at least one user-defined mode not being predetermined in terms of function, said plurality of modes controlling the execution of instructions within said instruction pipeline of said processor core in response to said at least one jump instruction word within said instruction set.

15. (Previously presented) The processor of Claim 14, wherein said plurality of jump delay slot modes comprises at least the following modes:

(i) executing a subsequent instruction within said pipeline under all circumstances;

(ii) executing a subsequent instruction within said pipeline only if jumping occurs; and

(iii) stalling the pipeline if jumping occurs.

16. (Previously presented) The processor of Claim 14, wherein said at least one jump instruction comprises a conditional branch instruction having an associated logical condition,

the execution of a jump to the address within said information storage device specified by said at least one conditional branch instruction being determined by said logical condition.

17. (Currently amended) A digital processor having at least one pipeline and an associated data storage device, wherein the execution of instructions within said at least one pipeline is controlled by the method comprising:

storing an instruction set within said data storage device, said instruction set comprising a plurality of instruction words, each of said instruction words comprising a plurality of data bits, at least one of said instruction words comprising a user-configurable branch instruction having a plurality of unique functional modes exclusively associated with respective ones of unique combinations of a plurality of mode control bits, ~~at least one user-defined mode,~~ said branch instruction directing branching to a first address within said data storage device;

assigning one of a plurality of values to each of said ~~data~~ mode control bits of said at least one branch instruction;

decoding said at least one branch instruction including said assigned values;

determining whether to execute an instruction within said pipeline in a stage preceding that of said at least one branch instruction based at least in part on said ~~one~~ assigned values;

branching to said first address based on said at least one branching instruction; and

performing, based at least in part on said act of decoding ~~decoding,~~ said assigned values, at least one other function dictated by ~~said at least one user-defined mode~~ the unique functional mode associated with said assigned values.

18. (Previously presented) The processor of Claim 17, wherein said data bits comprise binary (base 2) data.

19. (Previously presented) The method of Claim 17, wherein said at least one pipeline comprises at least a three stage instruction pipeline comprising instruction fetch, decode, and execute stages.

20. (Currently amended) A method of controlling program operation of a multi-stage pipelined digital processor having a data storage device in data communication therewith, comprising:

storing an instruction set within said data storage device, said instruction set comprising a plurality of instruction words, each of said instruction words comprising a plurality of data bits, at least one of said instruction words comprising a branch instruction directing branching to a first address within said data storage device based on a first parameter;

defining a plurality of jump delay slot modes comprising;

(i) executing a subsequent instruction under all circumstances;

(ii) executing a subsequent instruction only if jumping occurs; and

(iii) stalling the pipeline for one cycle if jumping occurs; ~~and~~

(iv) ~~stalling the pipeline for two or more cycles if jumping occurs~~;

assigning ~~at least one of~~ said plurality of jump modes to at least two of said data bits of said at least one branch instruction, wherein changing of any one or more of said at least two data bits will always specify a different functional mode;

decoding said at least one branch instruction including said at least two data bits; and

controlling said pipeline based at least in part on said at least two data bits and said first parameter.

21-22. (Cancelled)

23. (Cancelled)

24. (Cancelled)

25. (Previously Presented) The method of Claim 1, wherein at least one of said plurality of instruction words comprises an op-code and a plurality of fields, each of said fields comprising a plurality of bits, said at least one instruction word being encoded according to the method comprising:

associating a first of said fields with a first data source;

associating a second of said fields with a second data source; and

performing a logical operation using said first and second data sources as operands, said logical operation being specified by said op-code.

26. (Previously Presented) The method of Claim 1, further comprising generating a

long immediate constant using a single word instruction according to the method comprising:

providing an instruction word having an op-code and at least one short immediate value associated therewith, said at least one short immediate value comprising a plurality of bits;

selecting a portion of said plurality of bits of said at least one short immediate value;

shifting all of said bits of said at least one short immediate value using said op-code and only said portion of bits to produce a shifted immediate value; and

storing said shifted immediate value in a register.

27. (Previously Presented) The method of Claim 25, wherein said plurality of instruction words further comprises at least one instruction word having an op-code and at least one short immediate value associated therewith, said at least one short immediate value comprising a plurality of bits, said at least one instruction word with short immediate value being used to generate a long immediate constant according to the method comprising:

selecting a portion of said plurality of bits of said at least one short immediate value;

shifting all of said bits of said at least one short immediate value using said op-code and only said portion of bits to produce a shifted immediate value; and

storing said shifted immediate value in a register.

28. (Previously Presented) The method of Claim 27, wherein said at least one instruction word having a plurality of fields and said at least one instruction word having a short immediate value comprise the same instruction word(s).

29. (Previously Presented) The digital processor of Claim 14, wherein said at least one of said plurality of instruction words comprises an op-code and a plurality of fields, each of said fields comprising a plurality of bits, said at least one instruction word being encoded by:

associating a first of said fields with a first data source;

associating a second of said fields with a second data source; and

performing a logical operation using said first and second data sources as operands, said logical operation being specified by said op-code.

30. (Previously Presented) The digital processor of Claim 14, wherein said processor is further adapted to generate a long immediate constant using a single word instruction by:

providing an instruction word having an op-code and at least one short immediate value associated therewith, said at least one short immediate value comprising a plurality of bits;

selecting a portion of said plurality of bits of said at least one short immediate value;

shifting all of said bits of said at least one short immediate value using said op-code and only said portion of bits to produce a shifted immediate value; and

storing said shifted immediate value in a register.

31. (Previously Presented) The digital processor of Claim 29, wherein said plurality of instruction words further comprises at least one instruction word having an op-code and at least one short immediate value associated therewith, said at least one short immediate value comprising a plurality of bits, said at least one instruction word with short immediate value being used to generate a long immediate constant according to the method comprising:

selecting a portion of said plurality of bits of said at least one short immediate value;

shifting all of said bits of said at least one short immediate value using said op-code and only said portion of bits to produce a shifted immediate value; and

storing said shifted immediate value in a register.

32. (Previously Presented) The digital processor of Claim 31, wherein said at least one instruction word having a plurality of fields and said at least one instruction word having a short immediate value comprise the same instruction word(s).

33. (Cancelled)

34. (Cancelled)

35. (Cancelled)

36. (Cancelled)

37. (Currently amended) An extensible pipelined digital processor having an instruction set comprising a plurality of ~~basecase~~ instructions comprising a base instruction set and at least one extension instruction, at least one of said ~~basecase~~ base instruction set instructions ~~and~~ or extension instructions comprising a branch instruction having ~~at least one~~ a plurality of user-configurable ~~mode~~ modes ~~and a plurality of other modes~~ determined by a plurality of bits controlling the execution of at least one instruction in a delay slot following said

branch instruction within said pipeline~~.~~ , each of said modes being constrained to only one of a plurality of unique combinations of said plurality of bits

38. (Currently amended) An extensible pipelined digital processor having an instruction set comprising a plurality of ~~basecase~~ instructions forming a base instruction set and at least one extension instruction, at least one of said ~~basecase and extension~~ instructions comprising a branch instruction including two data bits defining four discrete modes controlling the execution of at least one instruction in a delay slot following said branch instruction within said pipeline;

wherein said execution is controlled without regard to a branch direction metric.

39. (Currently amended) An extensible pipelined digital processor having an instruction set, said processor comprising:

a ~~basecase~~ processor core configuration including a base instruction set; and

at least one user-configured extension instruction within said instruction set, said at least one extension instruction comprising a branch instruction having at least one ~~user-defined~~ mode ~~and a plurality of other modes~~ controlling the execution of at least one instruction in a delay slot following said branch instruction within said pipeline using a plurality of data bits, at least one of the particular combinations of data bits and the logical functions associated therewith being adapted for assignment by a user.

40. (Currently amended) An extensible pipelined digital processor having ~~basecase~~ base and extension instruction sets, at least one instruction within said ~~basecase~~ base set comprising a branch instruction having at least four discrete modes for controlling the execution of at least one instruction in a delay slot following said branch instruction within said pipeline, said processor comprising:

a ~~basecase~~ processor core configuration including said base instruction set; and

at least one user-customized extension instruction within said instruction set;

wherein each of said modes provides unique functionality with respect to the other three modes.

41. (Cancelled)

42. (Currently amended) The digital processor of Claim 40 [41], wherein first and second of said at least four ~~branch~~ modes implement one- and two-cycle stalls within said pipeline, respectively.

43. (Currently amended) The digital processor of Claim 40 [41], wherein at least one of said at least four ~~branch~~ modes ~~comprises a user-configurable mode~~ operates without respect to a branch displacement metric.

44. (New) The method of Claim 1, wherein said at least one user-definable mode comprises a non-jump or branch related mode.

45. (New) The method of Claim 1, wherein the operation of said at least one user-configurable or user-definable modes is not determined by a displacement value.

46. (New) The method of Claim 1, wherein the operation of said at least one user-configurable or user-definable modes is not dependent on an address calculation value.

47. (New) The method of Claim 44, wherein the operation of said at least one user-configurable or user-definable modes is not determined by a displacement value.

48. (New) The method of Claim 44, wherein the operation of said at least one user-configurable or user-definable modes is not dependent on an address calculation value.

49. (New) A digital processor comprising:

a processor core having a multistage instruction pipeline, said core being adapted to decode and execute an instruction set comprising a plurality of instruction words; and

an instruction set comprising a plurality of instruction words, at least one of said instruction words being a user-configurable jump instruction containing data defining a plurality of jump delay slot modes, said jump delay slot modes each being specified by the same portions of said data without reference to an address calculation metric, said plurality of modes controlling the execution of instructions within said instruction pipeline of said processor core in response to said at least one jump instruction word within said instruction set.

50. (New) A method of controlling program operation of a multi-stage pipelined digital processor, comprising:

storing an instruction set within said data storage device, said instruction set comprising a plurality of instruction words, each of said instruction words comprising a plurality of data bits, at least one of said instruction words comprising a branch instruction directing branching to a first address within said data storage device based on a first parameter;

defining a plurality of constrained jump delay slot functional modes comprising:

executing a subsequent instruction under all circumstances;

executing a subsequent instruction only if jumping occurs; and

stalling the pipeline for one cycle if jumping occurs;

assigning said plurality of jump modes to a plurality of said data bits of said at least one branch instruction, each of said functional modes being constrained to only one of a plurality of unique combinations of said plurality of bits;

decoding said at least one branch instruction including said plurality of data bits; and

controlling said pipeline based at least in part on said at least two data bits and said first parameter.

51. (New) A digital processor having at least one pipeline and adapted to interface with an associated data storage device and having an instruction disposed at least partly within said data storage device, said instruction set comprising a plurality of instruction words, at least one of said instruction words comprising a branch instruction having a plurality of functional modes associated with respective ones of combinations of a plurality of mode control bits, said branch instruction directing branching to a first address within said data storage device, wherein the execution of instructions within said at least one pipeline is controlled by the method comprising:

reading said mode control bits of said at least one branch instruction;

determining whether to execute an instruction within said pipeline in a stage different than that of said at least one branch instruction based at least in part on said read mode control bits and irrespective of the direction of said branching;

branching to said first address based on said at least one branching instruction; and

performing at least one other function dictated by the functional mode associated with said read bits.

52. (New) A digital processor having at least one pipeline and adapted to interface with an associated data storage device and having an instruction disposed at least partly within said data storage device, said instruction set comprising a plurality of instruction words, at least one of said instruction words comprising a branch instruction having a plurality of functional modes associated with respective ones of a plurality of combinations of a plurality of mode control bits, at least one of said plurality of combinations and the logical functions associated therewith being adapted for specification by a user, said branch instruction directing branching to a first address within said data storage device, wherein the execution of at least one instruction within said at least one pipeline is controlled by said mode control bits of said at least one branch instruction.